Approximation Algorithm for MAX-CUT

paper by
Michel X. Goemans and David P. Williamson
STOC 1994, JACM 1995

presented by
Chayant Tantipathananandh
MCS 541, Fall 2007

Outline

What is MAX-CUT?

- Given undirected graph $G = (V, E)$ with edge weights

$$w(u,v) = w(v,u) = \begin{cases} w_{uv} \in \mathbb{Z}^+ & \text{if } (u,v) \in E \\ 0 & \text{otherwise.} \end{cases}$$

- A cut S, $S \subseteq V$, partitions V into two sets S and $\overline{S}$.
- An edge (u, v) is in cut $(S, \overline{S})$ if
  - $u \in S$ and $v \in \overline{S}$, or
  - $u \in \overline{S}$ and $v \in S$.

Definition: MAX-CUT

Given an undirected graph, find a cut $(S, \overline{S})$ maximizing weight

$$w(S, \overline{S}) = \sum_{u \in S, v \in \overline{S}} w(u,v).$$

More about MAX-CUT

- MAX-CUT is NP-Complete

[Karp '72]

$$
\begin{aligned}
\text{SAT} &\leq_P \text{3SAT} \leq_P \text{CHROMATIC-NUMBER} \\
&\leq_P \text{EXACT-COVER} \leq_P \text{KNAPSACK} \\
&\leq_P \text{PARTITION} \leq_P \text{MAX-CUT}
\end{aligned}
$$

[Papadimitriou&Yannakakis '88]

$$
\text{SAT} \leq_P \text{3SAT} \leq_P \neq\text{-3SAT} \leq_P \text{MAX-CUT}
$$

Dealing with hard problems

- We want to find
  - *optimal* solution
  - on *all* inputs
  - *quickly* (in polynomial time)

Dealing with hard problems

- We want to find
  - *optimal* solution
  - on *all* inputs
  - *quickly* (in polynomial time)   relax $\Rightarrow$ superpolynomial time

Dealing with hard problems

- We want to find
  - *optimal* solution
  - on *all* inputs relax $\Rightarrow$ heuristics
  - *quickly* (in polynomial time) relax $\Rightarrow$ superpolynomial time

Dealing with hard problems

- We want to find
  - *optimal* solution relax ⇒ approximation
  - on *all* inputs relax ⇒ heuristics
  - *quickly* (in polynomial time) relax ⇒ superpolynomial time

Dealing with hard problems

- We want to find
  - *optimal* solution relax $\Rightarrow$ approximation
  - on *all* inputs relax $\Rightarrow$ heuristics
  - *quickly* (in polynomial time) relax $\Rightarrow$ superpolynomial time
- Constant approximations
  - $\frac{1}{2}$-approximation [Sahni & Gonzales '76].
  - α-approximation [Goemans & Williamson '94]

$$\alpha = \frac{2}{\pi} \min_{0 \le \theta \le \pi} \frac{\theta}{1 - \cos\theta} > 0.87856.$$

One Formulation

- Variables $x_i \in \{-1, 1\}$ for every vertex $i \in V$.
- Cut is $S = \{i : x_i = 1\}$ and $\overline{S} = \{i : x_i = -1\}$

Integer Quadratic Program (QP)

$$\text{maximize} \quad \frac{1}{2} \sum_{i<j} w_{ij}(1 - x_i x_j)$$
$$\text{subject to} \quad x_i^2 = 1 \qquad x_i \in \mathbb{R} \qquad \text{for every } i \in V$$

- Solving quadratic program is NP-Hard
(see Wikipedia for 0-1 integer program $\geq_P$ QP).
- All monomials have degree 0 or 2 (about this later).

Relaxation

- $x_i \in \{-1, 1\}$ is a vector lying on unit sphere S_1.
- Allow x_i to be a vector in higher-dimensional space.
  - $x_i \in S_1 \longrightarrow v_i \in S_n, n = |V|$
  - $x_i x_j \longrightarrow v_i \cdot v_j$

Vector Program (VP)

$$\text{maximize} \quad \frac{1}{2} \sum_{i<j} w_{ij}(1 - v_i \cdot v_j)$$

$$\text{subject to} \quad v_i \cdot v_i = 1 \qquad v_i \in \mathbb{R}^n \qquad \text{for every } i \in V$$

- Feasible solutions to QP are feasible solutions to VP with the same value.
- Objective function and constraints are linear in dot products. Solvable in polynomial time! (next slides).
- Problem: What is a cut? (later slides)

What is semidefinite matrix?

Definition

An $n \times n$ matrix $\mathbf{A}$ is positive semidefinite, written $\mathbf{A} \succeq 0$, when:

1. $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$ for all $\mathbf{x}$.
2. all eigenvalues of $\mathbf{A}$ are nonnegative real numbers.
3. $\mathbf{A}$ can be decomposed into matrix $\mathbf{B}$, $\mathbf{B}^T \mathbf{B} = \mathbf{A}$ (by Cholesky decomposition).

note: three statements are equivalent.

Frobenius Inner Product

Let $C = \{c_{ij}\}$ and $Y = \{y_{ij}\}$.

$$\begin{aligned} C \bullet Y &= c_{11}y_{11} + c_{12}y_{12} + \cdots + c_{ij}y_{ij} + \cdots + c_{kl}y_{kl} \\ &= \sum_{i=1}^{k} \sum_{j=1}^{l} c_{ij}y_{ij} \end{aligned}$$

$C \bullet Y$ is:

- an inner product of C and Y, or
- a linear function of variables y_{ij} with coefficients c_{ij}.

Semidefinite Programming

Definition

$$\begin{array}{ll} \text{maximize} & Z = C \bullet Y \\ \text{subject to} & A_i \bullet Y = b_i \quad \text{for } i = 1 \ldots m \\ & Y \succeq 0 \\ & Y \in \mathbb{R}^{n \times n} \end{array}$$

- Solvable to any additive error ϵ in polynomial time in input size and $\log \frac{1}{\epsilon}$ by, e.g.,
  - interior point method
  - ellipsoid algorithm

  i.e., can compute solution $\geq OPT_{SDP} - \epsilon$ in polynomial time.

Solving the Relaxation

- By definition, $B^T B \succeq 0$, so let $Y = B^T B$ and

$$B = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_n \\ | & | & & | \end{bmatrix}, \quad v_i \in \mathbb{R}^n.$$

- Replace $v_i \cdot v_j$ with y_{ij} to obtain...

Semidefinite Program (SDP)

$$\begin{array}{lll} \text{maximize} & \frac{1}{2}\sum_{i<j} w_{ij}(1 - y_{ij}) & \\ \text{subject to} & y_{ii} = 1 & \text{for every } i \in V \\ & \{y_{ij}\} \succeq 0 & \\ & y_{ij} \in \mathbb{R} & \end{array}$$

Approximation Algorithm for MAX-CUT

- Create semidefinite program
- Solve SDP for matrix Y in polynomial time.
- Perform incomplete Cholesky decomposition on Y to obtain

$$B = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_n \\ | & | & & | \end{bmatrix},$$

in polynomial time.

Approximation Algorithm for MAX-CUT

- Create semidefinite program
- Solve SDP for matrix Y in polynomial time.
- Perform incomplete Cholesky decomposition on Y to obtain

$$B = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_n \\ | & | & & | \end{bmatrix},$$

  in polynomial time.
- What is the maximum cut?

Approximation Algorithm for MAX-CUT

- Create semidefinite program
- Solve SDP for matrix Y in polynomial time.
- Perform incomplete Cholesky decomposition on Y to obtain

$$B = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_n \\ | & | & & | \end{bmatrix},$$

  in polynomial time.
- What is the maximum cut?
- Round $v_i \in S_n$ to a cut $x_i \in \{-1, 1\}$.

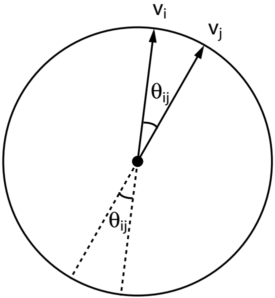
Randomized Rounding

- How about halving unit sphere S_n embedding v_i's?

Randomized Rounding

- How about halving unit sphere S_n embedding v_i's?
- Choose a hyperplane r uniformly at random:
  $S = \{i : v_i \cdot r \geq 0\}$ and $\overline{S} = \{i : v_i \cdot r < 0\}$.
- Expected weight of cut

$$E[W] = \sum_{i<j} w_{ij} \cdot \Pr[r \text{ separates } v_i, v_j] = \sum_{i<j} w_{ij} \cdot \frac{\theta_{ij}}{\pi}$$



Approximation Ratio

- How close to the maximum weight do we get?

$$\frac{E[W]}{OPT} \geq ?$$

Approximation Ratio

- How close to the maximum weight do we get?

$$\frac{E[W]}{OPT} \geq ?$$

- We know
  - $E[W] = \sum_{i<j} w_{ij} \cdot \frac{\theta_{ij}}{\pi}$
  - $OPT_{SDP} = \sum_{i<j} w_{ij} \left(\frac{1 - y_{ij}}{2} \right) = \sum_{i<j} w_{ij} \left(\frac{1 - \cos\theta_{ij}}{2} \right)$
  - $E[W] \leq OPT = OPT_{QP} \leq OPT_{SDP}$

Approximation Ratio

- How close to the maximum weight do we get?

$$\frac{E[W]}{OPT} \geq ?$$

- We know
  - $E[W] = \sum_{i<j} w_{ij} \cdot \frac{\theta_{ij}}{\pi}$
  - $OPT_{SDP} = \sum_{i<j} w_{ij} \left(\frac{1 - y_{ij}}{2} \right) = \sum_{i<j} w_{ij} \left(\frac{1 - \cos \theta_{ij}}{2} \right)$
  - $E[W] \leq OPT = OPT_{QP} \leq OPT_{SDP}$
- So

$$\frac{E[W]}{OPT} \geq \frac{E[W]}{OPT_{SDP}} = \frac{\sum_{i<j} w_{ij} \left(\frac{\theta_{ij}}{\pi} \right)}{\sum_{i<j} w_{ij} \left(\frac{1-\cos \theta_{ij}}{2} \right)} \geq \text{ratio!}$$

Approximation Ratio

- How close to the maximum weight do we get?

$$\frac{E[W]}{OPT} \geq ?$$

- We know
  - $E[W] = \sum_{i<j} w_{ij} \cdot \frac{\theta_{ij}}{\pi}$
  - $OPT_{SDP} = \sum_{i<j} w_{ij} \left(\frac{1 - y_{ij}}{2} \right) = \sum_{i<j} w_{ij} \left(\frac{1 - \cos \theta_{ij}}{2} \right)$
  - $E[W] \leq OPT = OPT_{QP} \leq OPT_{SDP}$
- So

$$\frac{E[W]}{OPT} \geq \frac{E[W]}{OPT_{SDP}} = \frac{\sum_{i<j} w_{ij} \left(\frac{\theta_{ij}}{\pi} \right)}{\sum_{i<j} w_{ij} \left(\frac{1 - \cos \theta_{ij}}{2} \right)} \geq \text{ratio!}$$

- Comparing them term-by-term.

Approximation Ratio (cont.)

... term-by-term comparison of

$$\frac{\theta}{\pi} \quad \text{to} \quad \frac{1 - \cos\theta}{2}$$

Approximation Ratio (cont.)

... term-by-term comparison of

$$\frac{\theta}{\pi} \quad \text{to} \quad \frac{1-\cos\theta}{2}$$

Algebra...

$$\begin{aligned} \frac{\theta}{\pi} &= \left(\frac{2}{\pi}\cdot\frac{\theta}{1-\cos\theta}\right)\left(\frac{1-\cos\theta}{2}\right) \\ &\geq \left(\frac{2}{\pi}\min_{0\leq\theta'\leq\pi}\frac{\theta'}{1-\cos\theta'}\right)\left(\frac{1-\cos\theta}{2}\right) \\ &= \alpha\left(\frac{1-\cos\theta}{2}\right). \end{aligned}$$

Approximation Ratio

Constant α is intentionally defined such that, for any $\theta \in [0, \pi]$,

$$\frac{\theta}{\pi} \geq \alpha \left(\frac{1 - \cos\theta}{2} \right).$$

Once again ...

$$\begin{aligned} E[W] &= \sum_{i<j} w_{ij} \cdot \Pr[v_i, v_j \text{ separated}] = \sum_{i<j} w_{ij} \left(\frac{\theta_{ij}}{\pi} \right) \\ &\geq \sum_{i<j} w_{ij} \cdot \alpha \left(\frac{1 - \cos\theta_{ij}}{2} \right) = \alpha \cdot OPT_{SDP} \\ &\geq \alpha \cdot OPT > 0.87856 \cdot OPT \end{aligned}$$

Conclusions

Conclusions:

- MAX-CUT problem, integer quadratic program formation
- Semidefinite program relaxation
- Random hyperplane rounding
- Approximation ratio $\alpha > 0.87856$

Related works:

- Integrality gap: Graphs that $E[W]$ is exactly $\alpha \cdot OPT$ [Karloff '97] [Feige&Schechtman '02]
- MAX 2SAT, MAX DICUT [Goemans&Williamson '94]
- Derandomization [Mahajan&Ramesh '95]
- CHROMATIC NUMBER (Graph Coloring) [Karger,Motwani,Sudan '94] [Arora&Chlamtac '06]
- Correlation Clustering [Swamy '94]

The End

Thank you.

Questions?

References

- Michel X. Goemans , David P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, JACM, v.42 n.6, Nov. 1995
- Vijay V. Vazirani, *Approximation Algorithms*, 2001.
- Howard Karloff, *How good is the Goemans-Williamson MAX CUT algorithm?*, STOC 1996.
- Uriel Feige, Gideon Schechtman, *On the Optimality of the Random Hyperplane Rounding Technique for MAX CUT*, Random Struct. Algorithms, 2002.
- Sanjeev Mahajan, H. Ramesh, *Derandomizing Semidefinite Programming Based Approximation Algorithms*, Random Struct. Algorithms, 2002.